What do we expect from an editor
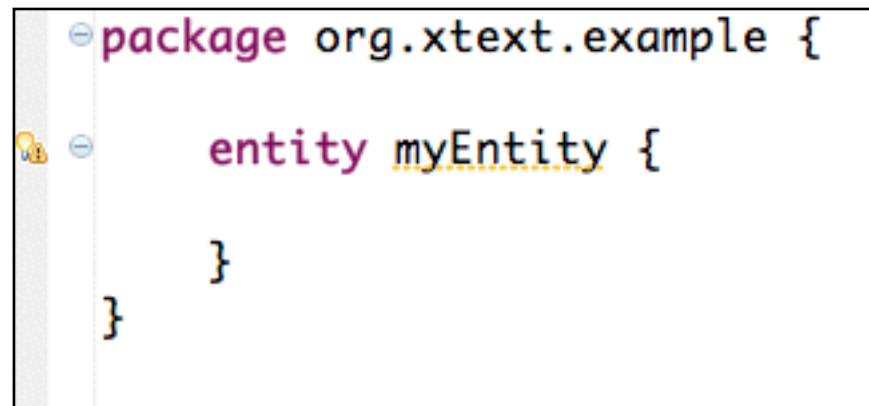
Syntaxhighlighting

```
package org.xtext.example {

    entity MyEntity {

    }
}
```

What do we expect from an editor

Syntaxhighlighting

```
package org.xtext.example {

    entity MyEntity {

    }
}
```

```
package org.xtext.example {

    entity myEntity {

    }
}
```
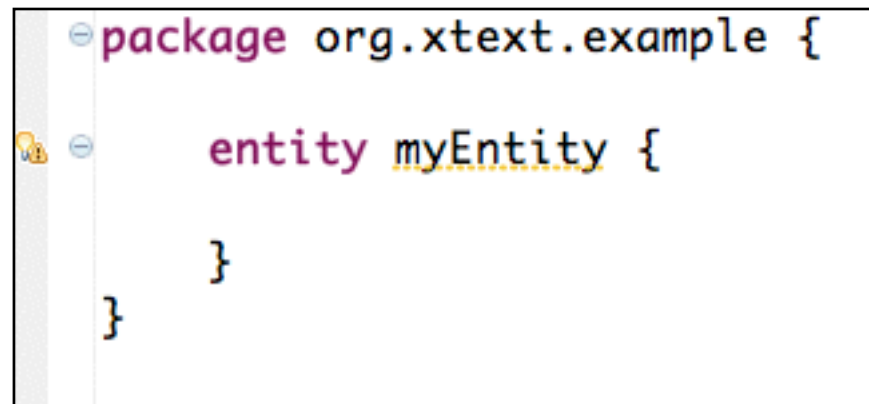
Validation / Quickfix / Annotations

What do we expect from an editor

Syntaxhighlighting

```
package org.xtext.example {

    entity MyEntity {

    }
}
```

Validation / Quickfix / Annotations

```
package org.xtext.example {

    entity myEntity {

    }
}
```

Contentassist

```
package org.xtext.example {

    entity myEntity {
        |
    }          op
}              ● Operation – template for an Operation
               ● Property – template for a Property
```

and much more …

What's the challenge?

Create an editor on a Composite

What's the challenge?

Create an editor on a Composite

Operate on a real or synthetic resource

What's the challenge?

Create an editor on a Composite

Operate on a real or synthetic resource

Edit only a part of the model

What's the challenge?

Create an editor on a Composite

Operate on a real or synthetic resource

Edit only a part of the model

Get the text for a specific object

What's the challenge?

Create an editor on a Composite

Operate on a real or synthetic resource

Edit only a part of the model

Get the text for a specific object

Synchronize back the changes

What's there in Xtext?

EmbeddedEditorFactory

```
G  org.eclipse.xtext.ui.editor.embedded.EmbeddedEditorFactory

Factory to create embedded editors for arbitrary resources. A typical usage pattern looks like this:

  EmbeddedEditorFactory editorFactory;

  EmbeddedEditor editor = editorFactory
      .newEditor(resourceProvider)
      .showErrorAndWarningAnnotations()
      .withParent(parentComposite);
  EmbeddedEditorModelAccess model = editor.createPartialEditor();
  // work with the model
```

```
EmbeddedEditor editor = editorFactory
      .newEditor(resourceProvider)
      .showErrorAndWarningAnnotations()
      .withParent(parentComposite);
EmbeddedEditorModelAccess model = editor.createPartialEditor();
```

Builder API

What about the resource?

```
public Builder newEditor(IEditedResourceProvider resourceProvider)
```

What about the resource?

```
public Builder newEditor(IEditedResourceProvider resourceProvider)


        public interface IEditedResourceProvider {
            XtextResource createResource();
        }
```

What about the resource?

```java
public Builder newEditor(IEditedResourceProvider resourceProvider)
```

```java
public interface IEditedResourceProvider {
    XtextResource createResource();
}
```

Resource could be empty or pre populated

What about the resource?

```
public Builder newEditor(IEditedResourceProvider resourceProvider)
```

```
public interface IEditedResourceProvider {
    XtextResource createResource();
}
```

Resource could be empty or pre populated

ResourceSet must be fully configured

What about the resource?

```java
@Inject
org.eclipse.xtext.ui.resource.IResourceSetProvider rsProvider

new IEditedResourceProvider() {

    @Override
    public XtextResource createResource() {

        XtextResourceSet resourceSet = (XtextResourceSet) resource.getResourceSet();
        final String SYNTHETIC_SCHEME = "synthetic";
        ResourceSet rs = rsProvider.get(((IJavaProject) resourceSet.getClasspathURIContext()).getProject());


        URI uri = URI.createURI(SYNTHETIC_SCHEME + ":/embeddedResource"+ "fileextension");
        XtextResource resource = (XtextResource) rs.createResource(uri);


        return resource;
    }
}
```

ResourceSet

What about the resource?

```java
@Inject
org.eclipse.xtext.ui.resource.IResourceSetProvider rsProvider

new IEditedResourceProvider() {

    @Override
    public XtextResource createResource() {

        XtextResourceSet resourceSet = (XtextResourceSet) resource.getResourceSet();
        final String SYNTHETIC_SCHEME = "synthetic";
        ResourceSet rs = rsProvider.get(((IJavaProject) resourceSet.getClasspathURIContext()).getProject());

        URI uri = URI.createURI(SYNTHETIC_SCHEME + ":/embeddedResource"+ "fileextension");
        XtextResource resource = (XtextResource) rs.createResource(uri);

        return resource;
    }
}
```

Resource

What about the resource?

```
@Inject
org.eclipse.xtext.ui.resource.IResourceSetProvider rsProvider

new IEditedResourceProvider() {

    @Override
    public XtextResource createResource() {

        XtextResourceSet resourceSet = (XtextResourceSet) resource.getResourceSet();
        final String SYNTHETIC_SCHEME = "synthetic";
        ResourceSet rs = rsProvider.get(((IJavaProject) resourceSet.getClasspathURIContext()).getProject());

        URI uri = URI.createURI(SYNTHETIC_SCHEME + ":/embeddedResource"+ "fileextension");
        XtextResource resource = (XtextResource) rs.createResource(uri);


        return resource;
    }
}
```

Resource

ResourceFactory identified by the fileextension

org.eclipse.xtext.resource.FileExtensionProvider

public String getPrimaryFileExtension()

Builder API

```
public Builder showErrorAndWarningAnnotations()
```

Show warnings and errors

Builder API

```
public Builder showErrorAndWarningAnnotations()
```

Show warnings and errors

```
public Builder showAnnotations(String... annotationTypes)
```

Builder API

```
public Builder showErrorAndWarningAnnotations()
```

Show warnings and errors

```
public Builder showAnnotations(String... annotationTypes)
```

```
public Builder processIssuesBy(IValidationIssueProcessor issueProcessor)
```

Create annotations (optional)

Builder API

```
public Builder showErrorAndWarningAnnotations()
```
Show warnings and errors

```
public Builder showAnnotations(String... annotationTypes)
```

```
public Builder processIssuesBy(IValidationIssueProcessor issueProcessor)
```
Create annotations (optional)

```
public Builder withResourceValidator(IResourceValidator resourceValidator)
```
Custom validator (optional)

Builder API

```
public Builder showErrorAndWarningAnnotations()
```

Show warnings and errors

```
public Builder showAnnotations(String... annotationTypes)
```

```
public Builder processIssuesBy(IValidationIssueProcessor issueProcessor)
```

Create annotations (optional)

```
public Builder withResourceValidator(IResourceValidator resourceValidator)
```

Custom validator (optional)

```
public Builder readOnly()
```

Builder API

```java
public Builder showErrorAndWarningAnnotations()
```

Show warnings and errors

```java
public Builder showAnnotations(String... annotationTypes)
```

```java
public Builder processIssuesBy(IValidationIssueProcessor issueProcessor)
```

Create annotations (optional)

```java
public Builder withResourceValidator(IResourceValidator resourceValidator)
```

Custom validator (optional)

```java
public Builder readOnly()
```

```java
public EmbeddedEditor withParent(final Composite parent)
```

Edit only a part of the model

```java
public EmbeddedEditorModelAccess createPartialEditor(
                    String prefix, String editablePart, String suffix,
                    boolean insertLineBreaks)
```
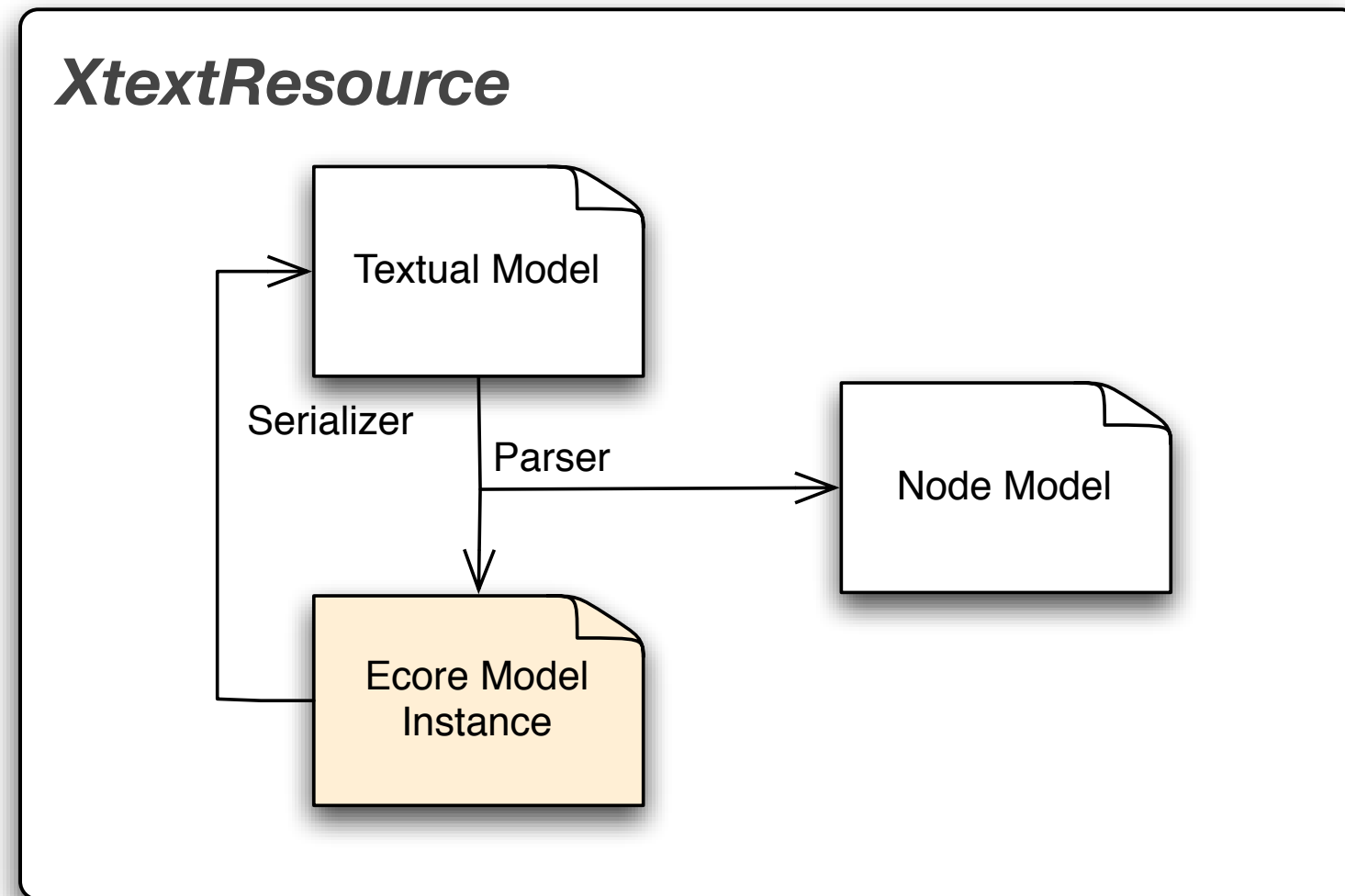
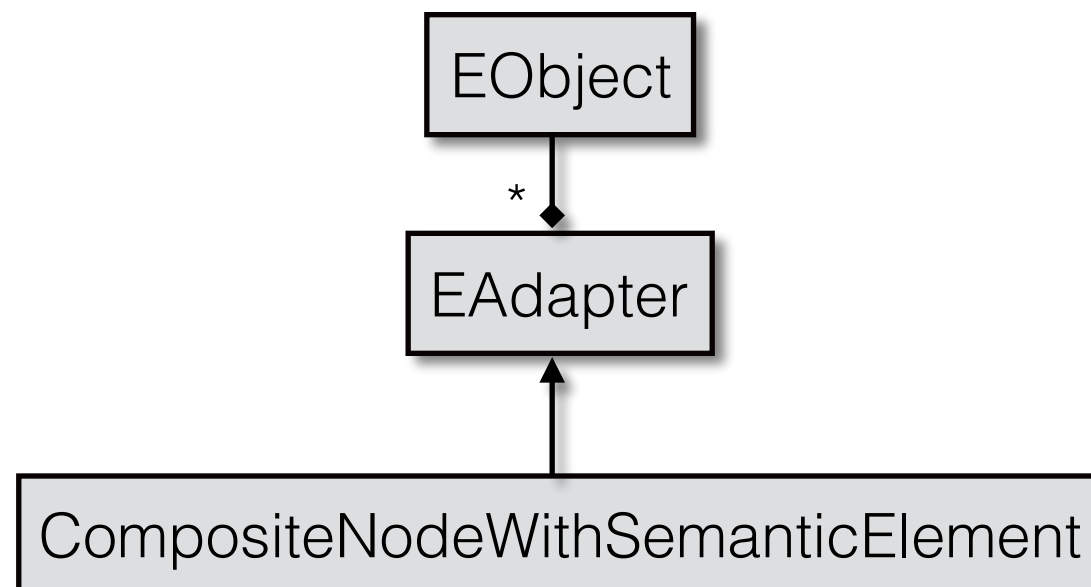Prefix - non visible part

editablePart - visible part
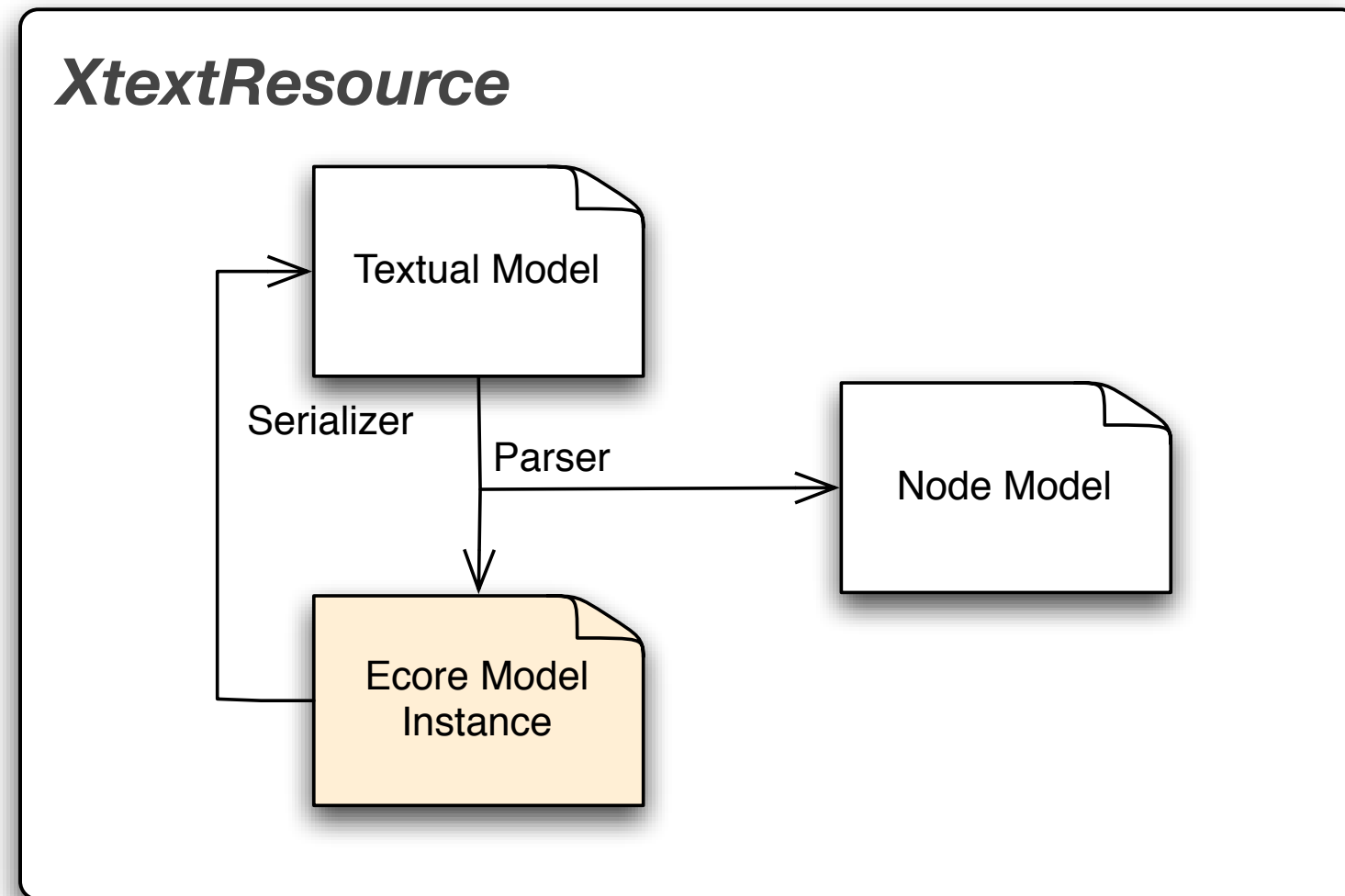
suffix - non visible part

insertLineBreaks - linebreaks between the parts

How to get the editablePart?
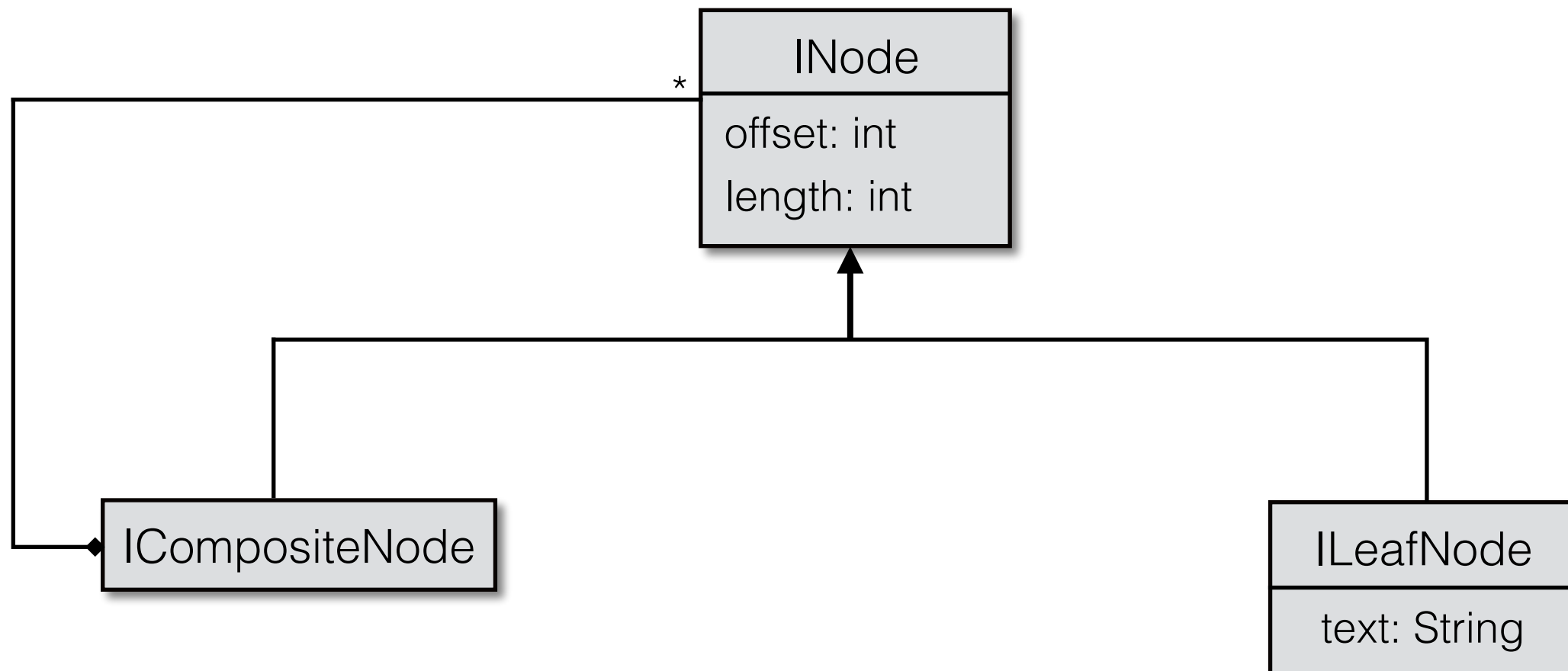
How to get the editablePart?

XtextResource

Textual Model

Serializer

Parser → Node Model

Ecore Model Instance

EObject

*

EAdapter

CompositeNodeWithSemanticElement

How to get the editablePart?

How to get the editablePart?

```
         ┌─────────────────┐
    *    │      INode      │
┌────────│─────────────────│
│        │ offset: int     │
│        │ length: int     │
│        └─────────────────┘
│                  △
│        ┌─────────┴──────────────┐
│ ┌──────┴──────────┐   ┌─────────┴────────┐
└▶│ ICompositeNode  │   │    ILeafNode     │
  └─────────────────┘   │──────────────────│
                        │   text: String   │
                        └──────────────────┘
```
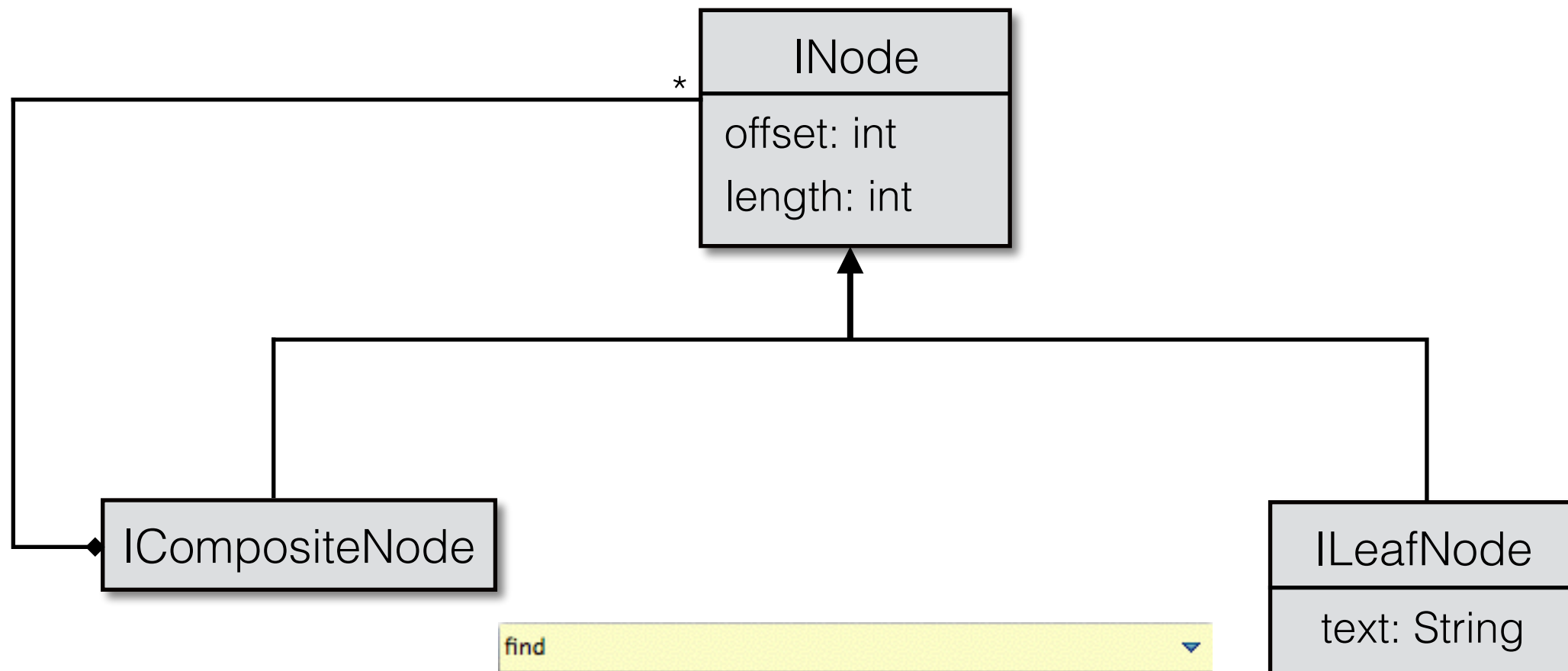
find   ▼

▼ⓖ **NodeModelUtils**
  ● ˢ findLeafNodeAtOffset(INode, int) : ILeafNode
  ● ˢ findNodesForFeature(EObject, EStructuralFeature) : List
  ● ˢ findActualNodeFor(EObject) : ICompositeNode
  ● ˢ findActualSemanticObjectFor(INode) : EObject

Press "⌘O" to show inherited members

Synchronize back the changes

Depends on the use case

Synchronize back the changes

Depends on the use case

Simple String edited with tooling and written
back to the model as a String

Synchronize back the changes

Depends on the use case

Simple String edited with tooling and written
back to the model as a String

Complete model edited and editable part
should be updated in resource

Synchronize back the changes

org.eclipse.xtext.resource.XtextResource.update(int, int, String)

```java
public void update(int offset, int replacedTextLength, String newText)
```

Synchronize back the changes

```
org.eclipse.xtext.resource.XtextResource.update(int, int, String)
```

```
public void update(int offset, int replacedTextLength, String newText)
```

We already know offset and length from the Node

Update the embedded editor

```java
public EmbeddedEditorModelAccess createPartialEditor(
                    String prefix, String editablePart, String suffix,
                    boolean insertLineBreaks)
```

org.eclipse.xtext.ui.editor.embedded.EmbeddedEditorModelAccess

```java
public void updateModel(String prefix, String editablePart, String suffix)
```

DEMO

Thank you!

XteXtCON